club commodore

n°8

```
AD1E; Perform [NEXT]
AD78; Type match check
AD9E; Evaluate expression
AEA8; Constant - pi
AEF1; Evaluate within brackets
AEF7;    °)°
AEFF;    comma..
AF08; Syntax error
AF14; Check range
AF28; Search for variable
AFA7; Setup FN reference
AFE6; Perform [OR]
AFE9; Perform [AND]
B016; Compare
B081; Perform [DIM]
B08B; Locate variable
B113; Check alphabetic
B11D; Create variable
B194; Array pointer subrtine
B1A5; Value 32768
B1B2; Float-fixed
B1D1; Set up array
B245; °bad subscript°
B248; °illegal quantity°
B34C; Compute array size
B37D; Perform [FRE]
B391; Fix-float
B39E; Perform [POS]
B3A6; Check direct
B3B3; Perform [DEF]
B3E1; Check fn syntax
B3F4; Perform [FN]
B465; Perform [STR$]
B475; Calculate string vector
B487; Set up string
B4F4; Make room for string
B526; Garbage collection
B5BD; Cneck salvageability
B606; Collect string
B63D; Concatenate
B67A; Build string to memory
B6A3; Discard unwanted string
B6DB; Clean descriptor stack
B6EC; Perform [CHR$]
B700; Perform [LEFT$]
B72C; Perform [RIGHT$]
B737; Perform [MID$]
B761; Pull string parameters
B77C; Perforr [LEN]
B782; Exit string-mode
B78B; Perform [ASC]
B79B; Input byte paramter
B7AD; Perform [VAL]
B7EB; Parameters for POKE/WAIT
B7F7; Float-fixed
B80D; Perform [PEEK]
B824; Perform [POKE]
B82D; Perform [WAIT]

B849; Add 0.5
B850; Subtract-from
B853; Perform [subtract]
B86A; Perform [add]
B947; Complement FAC#1
B97E; °overflow°
B983; Multiply by zero byte
B9EA; Perform [LOG]
BA2B; Perform [multiply]
BA59; Multiply-a-bit
BA8C; Memory to FAC#2
BAB7; Adjust FAC#1/#2
BAD4; Underflow/overflow
BAE2; Multiply by 10
BAF9; +10 in floating pt
BAFE; Divide by 10
BB12; Perform [divide]
BBA2; Memory to FAC#1
BBC7; FAC#1 to memory
BBFC; FAC#2 to FAC#1
BC0C; FAC#1 to FAC#2
BC1B; Round FAC#1
BC2B; Get sign
BC39; Perform [SGN]
BC58; Perform [ABS]
BC5B; Compare FAC#1 to mem
BC9B; Float-fixed
BCCC; Perform [int]
BCF3; String to FAC
BD7E; Get ascii digit
BDC2; Print °IN..°
BDCD; Print line number
BDDD; Float to ascii
BF16; Decimal constants
BF3A; TI constants
BF71; Perform [SQR]
BF7B; Perform [power]
BFB4; Perform [negative]
BFED; Perform [EXP]
E043; Series eval 1
E059; Series eval 2
E097; Perform [RND]
E0f9; ?? breakpoints ??
E12A; Perform [SYS]
E156; Perform [SAVE]
E165; Perform [VERIFY]
E168; Perform [LOAD]
E1BE; Perform [OPEN]
E1C7; Perform [CLOSE]
E1D4; Parameters for LOAD/SAVE
E206; Check default parameters
E20E; Check for comma
E219; Parameters for open/close
E264; Perform [COS]
E26B; Perform [SIN]
E2b4; Perform [TAN]
E30E; Perform [ATN]
E37B; Warm restart

E394; Initialize
E3A2; CHRGET for zero page
E3BF; Initialize Basic
E447; Vectors for $300
E453; Initialize vectors
E45F; Power-up message
E500; Get I/O address
E505; Get screen size
E50A; Put/get row/column
E518; Initialize  I/O
E544; Clear screen
E566; Home cursor
E56C; Set screen pointers
E5A0; Set I/O defaults
E5B4; Input from keyboard
E632; Input from screen
E684; Quote test
E691; Setup screen print
E6B6; Advance cursor
E6ED; Retreat cursor
E701; Back into previous line
E716; Output to screen
E87C; Go to next line
E891; Perform ›return‹
E8A1; Check line decrement
E8B3; Check line increment
E8CB; Set color code
E8DA; Color code table
E8EA; Scroll screen
E965; Open space on screen
E9C8; Move a screen line
E9E0; Synchronize color transfer
E9F0; Set start-of-line
E9FF; Clear screen line
EA13; Print to screen
EA24; Synchronize color pointer
EA31; Interrupt - clock etc
EA87; Read keyboard
EB79; Keyboard select vectors
EB81; Keyboard 1 - unshifted
EBC2; Keyboard 2 - shifted
EC03; Keyboard 3 - °comm°
EC44; Graphics/text contrl
EC4F; Set graphics/text mode
EC78; Keyboard 4
ECB9; Video chip setup
ECE7; Shift/run equivalent
ECF0; Screen ln address low
ED09; Send °talk°
ED0C; Send °listen°
ED40; Send to serial bus
EDB2; Serial timeout
EDB9; Send listen SA
EDBE; Clear ATN
EDC7; Send talk SA
EDCC; Wait for clock
EDDD; Send serial deferred
EDEF; Send °untalk°

EDFE; Send °unlisten°
EE13; Receive from serial bus
EE85; Serial clock on
EE8E; Serial clock off
EE97; Serial output °1°
EEA0; Serial output °0°
EEA9; Get serial in & clock
EEB3; Delay 1 ms
EEBB; RS-232 send
EF06; Send new RS-232 byte
EF2E; No-DSR error
EF31; No-CTS error
EF3B; Disable timer
EF4A; Compute bit count
F8D0; Check tape stop
F8E2; Set read timing
F92C; Read tape bits
FA60; Store tape chars
FB8E; Reset pointer
FB97; New character setup
FBA6; Send transition to tape
FBC8; Write data to tape
FBCD; IRQ entry point
FC57; Write tape leader
FC93; Restore normal IRQ
FCB8; Set IRQ vector
FCCA; Kill tape motor
FCD1; Check r/w pointer
FCDB; Bump r/w pointer
FD50; Initialize system constnts
FCE2; Power reset entry
FD02; Check 8-rom
FD10; 8-rom mask
FD15; Kernal reset
FD1A; Kernal move
FD30; Vectors
FD9B; IRQ vectors
FDA3; Initialize I/O
FDDD; Enable timer
FDF9; Save filename data
FE00; Save file details
FE07; Get status
FE18; Flag status
FE1C; Set status
FE21; Set timeout
FE25; Read/set top of memory
FE27; Read top of memory
FE2D; Set top of memory
FE34; Read/set bottom of memory
FE43; NMI entry
FE66; Warm start
FEB6; Reset IRQ & exit
FEBC; Interrupt exit
FEC2; RS-232 timing table
FED6; NMI RS-232 in
FF07; NMI RS-232 out
FF43; Fake IRQ
FF48; IRQ entry
FF81; Jumbo jump table
```

EDITORIAL

algunas de las novedades que COMMODORE tiene anunciadas

Nuestra Revista, en su habitual esfuerzo por facilitar a sus lectores una información lo más fresca posible, ha enviado a la recientemente celebrada feria de HANNOVER a dos espías con el fin de averiguar qué novedades está preparando COMMODORE para el próximo futuro. Una vez sometidos nuestros enviados a un exhaustivo tercer grado han confesado haber visto lo siguiente:

PRIMERO: *El lector de cintas de cassette tiene un nuevo diseño (hemos conseguido arrancarles hasta una fotografía) y aunque sólo podemos evaluar los cambios externos creemos que la mejora es sustancial: mejor visibilidad del cuentavueltas y de la cinta y más teclas de actuación situadas en posiciones diferentes de las antiguas indudablemente para amenizar el trabajo de los que estamos acostumbrados al modelo antiguo.*

SEGUNDO: *Parece ser que también existe (los servicios de contraespionaje han conseguido interceptar los microfilms) una nueva versión del COMMODORE 64 ¡¡PORTÁTIL!! es decir, integrado en una caja con una pantalla en color de 5 pulgadas y una unidad de disco y espacio para otra.*

TERCERO: *Aparece también un "plotter" (una impresora que permite dibujar directamente) de cuatro colores que —siguiendo la política clásica de COMMODORE— tendrá un precio de risa. El sistema de dibujo es por bolígrafo, su resolución es de 0,2 mm. Este aparato —cuyo modelo es el 1520, aseguran nuestros hombres en HANNOVER— es capaz de escribir además de dibujar un máximo de 80 caracteres por línea en diferentes tamaños y orientaciones, tiene un tamaño reducido y puede sustituir a la impresora en aquellas aplicaciones que requieran el uso frecuente de gráficos.*

Como colofón a esta serie de novedades hablaremos de una existente en nuestro país: la unidad de disco 1541 que sustituye a la del VIC y que sirve directamente para el COMMODORE 64, para aquellos que posean un monofloppy 1540 y necesiten convertirlo a 1541 (el del COMMODORE 64) se está empezando a comercializar una ROM que realiza esta función.

En cuanto a nuestra Revista los lectores habituales habrán observado que hemos vuelto a aumentar el número de páginas (en la edición para nuestros suscriptores) por supuesto sin modificar el precio. En cuanto al contenido estamos preparando algunas nuevas secciones que empezarán a aparecer dentro de muy poco. ■

VENTANA CBM

modo de utilizar variables del BASIC desde código máquina

por **JOSEP TINTO ESPELT**

Ante todo recomiendo que se repase el contenido de la VENTANA CBM del número seis de nuestra Revista con el fin de que todo lo relacionado con la estructura de las variables en la memoria del ordenador quede totalmente claro.

Es frecuente la necesidad, al desarrollar subrutinas de código máquina para ser llamadas desde el BASIC, que se presente la conveniencia de recoger datos de una variable tanto numérica como alfanumérica desde la subrutina propiamente dicha. El caso contrario es también frecuente (afectar una variable con un nuevo contenido desde la propia subrutina).

(continúa en la pág. siguiente)

VENTANA CBM

modo de utilizar variables del BASIC desde código máquina

(viene de la pág. anterior)

PUNTEROS UTILIZADOS EN LOS EJEMPLOS

Etiqueta	Decimal	hexa	Descripción
flag	2	$02	flag utilizado en las rutinas.
COUNT	5	$05	contador general (número de índices).
DIMFLG	6	$06	dim flag.
VALTYP	7	$07	tipo de variable 0=numérica 255=alfa
INTFL	8	$08	flag para tipo entero,
VARTAB	42,43	$2A,2B	puntero tabla de variables.
ARYTAB	44,45	$2C,2D	puntero tabla de matrices.
STREND	46,47	$2E,2F	puntero fin de tablas.
FRETOP	48,49	$30,31	puntero principio strings.
FRESPC	50,51	$32,33	puntero tope de espacio strings.
MENSIZ	52,53	$34,35	puntero posición máx. memoria útil,
VARNAM	66,67	$42,43	nombre variable en curso.
VARPNT	68,69	$44,45	puntero de dir. en tabla.
FAC	94,99	$5E,63	acumulador # 1.
ARG	102,107	$66,6B	acumulador # 2.
STRNG2	110,111	$6E,6F	índice.
índice	214,215	$D6,D7	índice de utilidad.

SUBRUTINAS UTILIZABLES BASIC 4.0

Etiqueta	Decimal	hexa	Descripción
bvar	49543	$C187	búsqueda de variable.
barr	49987	$C343	búsqueda de matriz.
GARBA2	50794	$C66A	recogida de basura (garbage).
CONUPK	52162	$CBC2	carga ARG con memoria index. (a,y).
MOVFM	52440	$CCD8	carga FAC con memoria index. (a,y).
MOVMF	52490	$CD0A	carga memoria index. (x,y) con FAC.
MOVFA	52530	$CD32	carga FAC con contenido de ARG.
MOVAF	52546	$CD42	carga ARG con contenido de FAC.

SUBRUTINA DE BÚSQUEDA DE VARIABLE NUMÉRICA

```
BUVN          ; modo de buscar var. num.
lda # 'a      ; nombre variable.
ldx # 'b      ; segundo carácter.
(ldx # 0)     ; sin segundo carácter.
sta varnam    ; primer carácter a varnam
stx varnam+1  ; segundo carácter.
lda # 0
sta VALTYP    ; tipo de variable.
sta INTFL     ; flag de entero.
jsr bvar      ; subrutina del basic.
jmp lflp      ; carga en FAC.
```

SUBRUTINA DE BÚSQUEDA DE VARIABLE ALFANUMÉRICA

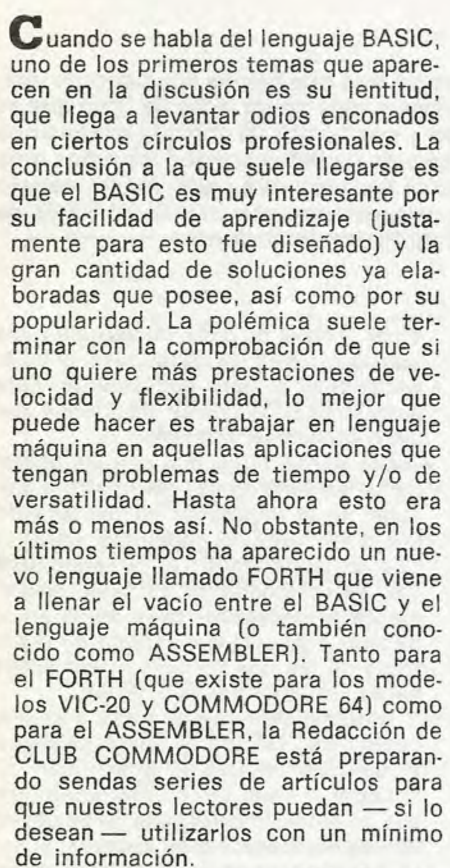
```
BVAL          ; modo de buscar una var. alfa.
lda # 'a      ; primer carácter nombre var.
ldx # 'b      ; segundo carácter nombre.
(ldx # 0)     ; sin segundo carácter.
ora # 128     ; distintivo alfanum. $
lda # 255
sta VALTYP    ; tipo var. alfanumérica.
lda # 0
sta INTFL     ; flag entero
jmp bvar      ; subrutina del basic.
```

NOTA: El puntero de página cero VARPNT contendrá, a la salida, la dirección de la tabla de variables en el byte que contiene la longitud del «string». La dirección del «string» estará en los dos bytes siguientes.

Continuará en el próximo número.

BASIC

cómo obtener programas más rápidos en BASIC



por **P. MASATS**

Cuando se habla del lenguaje BASIC, uno de los primeros temas que aparecen en la discusión es su lentitud, que llega a levantar odios enconados en ciertos círculos profesionales. La conclusión a la que suele llegarse es que el BASIC es muy interesante por su facilidad de aprendizaje (justamente para esto fue diseñado) y la gran cantidad de soluciones ya elaboradas que posee, así como por su popularidad. La polémica suele terminar con la comprobación de que si uno quiere más prestaciones de velocidad y flexibilidad, lo mejor que puede hacer es trabajar en lenguaje máquina en aquellas aplicaciones que tengan problemas de tiempo y/o de versatilidad. Hasta ahora esto era más o menos así. No obstante, en los últimos tiempos ha aparecido un nuevo lenguaje llamado FORTH que viene a llenar el vacío entre el BASIC y el lenguaje máquina (o también conocido como ASSEMBLER). Tanto para el FORTH (que existe para los modelos VIC-20 y COMMODORE 64) como para el ASSEMBLER, la Redacción de CLUB COMMODORE está preparando sendas series de artículos para que nuestros lectores puedan — si lo desean — utilizarlos con un mínimo de información.

Pero volvamos al BASIC. En este artículo se darán una serie de indicaciones sobre la manera de obtener la máxima velocidad de ejecución de los programas escritos en este lenguaje, mediante la racionalización del redactado de los mismos. El único inconveniente de estos métodos de optimización es que los programas serán menos «legibles»; es decir, que una persona que lea un listado encontrará más dificultades en saber qué hace el programa y cómo lo hace. Y no sólo otra persona tendrá dificultades para entender el programa que ha realizado Ud. sino que Ud. mismo, al cabo de seis meses de haber escrito un programa, tendrá que luchar encarnizadamente para recordar el proceso que veía tan claro. De todas maneras, no se preocupe; a nosotros nos pasa lo mismo aunque no utilicemos técnicas de optimización y es que BUG nunca descansa.

Para entender las razones de algunos de los trucos que se explican es conveniente consultar la serie de artículos que se han publicado en nuestra sección VENTANA CBM en los números 3, 4, 5 y 6 sobre la estructura de los programas BASIC y el almacenamiento de variables. Algunas de las técnicas de optimización son las siguientes:

1. Escriba sus subrutinas al principio de su programa y ponga una línea antes de éstas con un GOTO que nos remita al principio real del programa. Cuando el intérprete de BASIC encuentra una instrucción GOSUB ..., comienza a buscar la subrutina al principio del programa. Si éstas están al principio las encuentra con más rapidez. Ejemplo:

```
  1 GOTO 100
 10 REM PRINCIPIO DE SUBRUTINAS
 99 RETURN: REM FIN DE SUBRUTINAS
100 REM COMIENZO DEL PROGRAMA PROPIAMENTE DICHO
```

2. Evite los espacios. Los espacios son caracteres que deben ser ignorados y esto toma tiempo en la interpretación. Ejemplo:

```
1 PRINT 7 + 9: REM ES MÁS LENTO QUE:
1 PRINT7+9:REM QUE ES MÁS RÁPIDO
```

3. No incluya REMs. Ocurre lo mismo que con los espacios.

4. Utilice nombres de variables de una sola letra. El intérprete se toma su tiempo para leer el carácter extra; además, se emplea más memoria.

5. Incluya el máximo de instrucciones en una misma línea utilizando separadores ":". Esto reduce el número de líneas del programa, acelerando los GOTOs. Cuando se ejecuta una instrucción de salto como GOTO o GOSUB, el intérprete empieza su búsqueda por la primera línea y continúa una por una buscando la que necesita. Esta precaución también ahorra memoria.

6. Numere las líneas de su programa de una en una. Esto reduce los dígitos que han de leerse durante un GOTO o un GOSUB y de paso ahorra memoria.

7. No utilice variables enteras (ej.: A%). Cuando el intérprete de BASIC encuentra una variable entera tiene que convertirla a coma flotante para poder manejarla. Cuando se debe manejar una instrucción del tipo A%=B%+C%, el BASIC debe convertir B% y C% a coma flotante, realizar la suma y reconvertir el resultado a entero para asignarlo a la variable A%. Todo esto consume tiempo, aparte del carácter extra (%) que se debe almacenar y manejar.

8. Utilice variables como constantes en su programa. El intérprete debe convertir las constantes a coma

(termina en la pág. siguiente)

cómo obtener programas más rápidos en BASIC

(viene de la pág. anterior)

flotante para poder manejarlas. Si las constantes ya están convertidas (en las variables) habrá un ahorro de tiempo. Ejemplo:

```
10 POKE36879,8:REM ES MÁS LENTO
   QUE
10 POKEQ,J:REM QUE ES MÁS RÁPI-
   DO
```

9. Utilice instrucciones PRINT en vez de POKEs en pantalla siempre que sea posible. La instrucción PRINT establece el color del carácter además del carácter en sí lo cual requería dos POKEs.

10. No escriba IF ... THENGOTO ... El GOTO es innecesario.

11. Atribuya un valor inicial a todas las variables al principio de su programa, antes de que se dimensione ninguna matriz o se definan las cadenas alfanuméricas. Cuando se crean nuevas variables, el intérprete debe mover las matrices y las cadenas para «hacer sitio», lo cual consume tiempo (ver los artículos citados).

12. Ponga las variables utilizadas con más frecuencia las primeras en las instrucciones de valor inicial. El BASIC encontrará los valores de estas variables antes.

13. No use el nombre de la variable en una instrucción NEXT. Cuando se encuentra esta instrucción, el intérprete comprueba si el nombre es correcto. Si sólo se especifica NEXT el BASIC asume que es correcto y continúa con la ejecución.

14. Utilice rutinas en lenguaje máquina en los puntos críticos. Hay un límite de velocidad que el BASIC no puede superar El lenguaje máquina se ejecuta a mucha mayor velocidad. Cortas rutinas en ASSEMBLER en los puntos críticos pueden incrementar notablemente la velocidad del programa, sin complicarlo excesivamente. ■

COLABORACIONES

1. programa para e de la ley de Ohm
2. un tren de jugue la pantalla de t

Hemos recibido estos dos programas que hemos considerado interesantes para el VIC-20. El primero trata sobre la ley de Ohm que es la base de toda la electricidad y electrónica tal como la estudiamos y el segundo es un sencillo tren de juguete para que los niños muy pequeños pierdan el respeto al teclado (¡si es que tienen respeto por algo!). Sus autores son, respectivamente: JOAQUÍN SANZ de Barcelona y F. CASTANYER de Palma de Mallorca.

JOAQUÍN SANZ nos explica:

El presente programa está basado en las fórmulas:

$$V=R*I \qquad P=V*I$$

Y está creado con el fin de, en función de cualquier par de parámetros, realizar el cálculo de cualquiera de los restantes y, si se desea, también del cuarto.

Con la combinación de las diferentes formas en que pueden verse las dos fórmulas antes citadas obtenemos un rosario de 12 con el que podemos calcular cualquier valor en base a un par de los demás, sin necesidad de realizar cálculos y despejes previos. Ésta es fundamentalmente la ventaja que persigue este programa.

FUNCIONAMIENTO

Las líneas 5-90 imprimen la cabecera del programa, junto a una somera explicación de lo que será el mismo, y se exponen los parámetros con los que trabaja el programa.

Las líneas siguientes, de la 100 a la 160, solicitan al operador el parámetro a calcular, pasando a la subrutina de selección de las fórmulas que veremos luego. En el caso de un error al darle un dato, pulsar una tecla que no entre en las solicitadas, el ordenador pide el dato nuevamente. Este paso se da en todas las elecciones que debe realizar el operador.

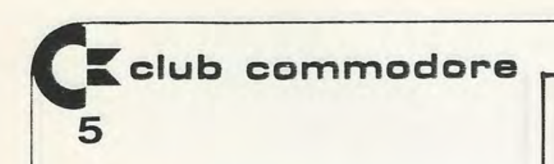
Las líneas 175 a 500 son las encargadas en función del parámetro a calcular de la impresión en pantalla de las tres fórmulas posibles para el cálculo del parámetro en cuestión. Una vez seleccionada la fórmula conveniente, el ordenador pasa a una serie de subrutinas (4) que solicitarán la entrada de las dos variables necesarias para el cálculo (líneas 510-580).

Automáticamente se pasa a la ejecución del cálculo y posteriormente el operador tendrá la posibilidad de solicitar el valor del cuarto parámetro que no entra en la fórmula. Si se desea conocerlo aparecerá en pantalla, si no, se pasa a la posibilidad de realizar un nuevo cálculo, o bien finalizar el programa.

Al cálculo del cuarto parámetro corresponden las líneas 870-1260, en las que va incluida la decisión de calcularlo o no.

Las líneas 1270-1310 permiten la continuación del cálculo o el fin del programa.

Por lo que respecta a los cálculos, las líneas se corresponden así:



l estudio

te en

elevisor

RESISTENCIA: 590-650
INTENSIDAD: 660-720
TENSIÓN: 730-790
POTENCIA: 800-860

```
READY.

10 PRINT"[ CLR ]";SPC(4)"**********
****"

20 PRINTSPC(3)"**4-PARAMETROS**"

40 PRINTSPC(4)"**************"

50 PRINT:PRINT"J.SANZ,22/03/83"

60 PRINT:PRINTSPC(1)"ESTE PROGRAMA
CALCULA LOS 4 PARAMETROS ELE-MENTAL
ES DE UN CIRCUI-"

65 PRINT"TO ELECTRICO SIMPLE";" EST
OS PARAMETROS SON:"

70 PRINT:PRINT"R-RESISTENCIA(OHM)"

75 PRINT"I-INTENSIDAD(AMPER)"

80 PRINT"V-TENSION(VOLT)"

90 PRINT"P-POTENCIA(WATT)"

95 REM SELECCION DEL PARAMETRO Y FO
RMULA PARA SU CALCULO

100 PRINT:PRINT"QUE VALOR DESEA CAL
CU-LAR?"

110 INPUT"R/I/V/P";A$

120 IF A$="R" THEN GOSUB 170:GOTO 1
90

130 IF A$="I" THEN GOSUB 170:GOTO27
0

140 IFA$="V"THEN GOSUB170:GOTO350

150 IFA$="P"THENGOSUB170:GOTO430

160 GOTO110

170 PRINT"[ CLR ]":PRINT"CUAL DE LA
S FORMULAS  SIGUIENTES LE CONVIENE?
"

180 RETURN

185 REM FORMULAS PARA EL CALCULO

190 PRINT:PRINT"1- R=V/I"

200 PRINT"2- R=P/(I↑2)"

210 PRINT"3- R=V↑2/P"

220 INPUT"1/2/3";B$

225 PRINT:PRINT

230 IFB$="1"THENGOSUB550:GOSUB530:G
OTO600

240 IFB$="2"THEN GOSUB570:GOSUB530:
GOTO620

250 IFB$="3" THEN GOSUB550:GOSUB570
:GOTO640

260 GOTO190

270 PRINT:PRINT"1- I=V/R"

280 PRINT"2- I=P/V"

290 PRINT"3- I=SQR(P/R)"

300 INPUT"1/2/3";C$

305 PRINT:PRINT

310 IFC$="1"THENGOSUB550:GOSUB510:G
OTO670

320 IFC$="2"THENGOSUB570:GOSUB550:G
OTO690

330 IFC$="3"THENGOSUB570:GOSUB510:G
OTO710

340 GOTO270

350 PRINT:PRINT"1- V=R*I"

360 PRINT"2- V=P/I"

370 PRINT"3- V=R*SQR(P/R)"

380 INPUTD$

385 PRINT:PRINT

390 IFD$="1"THEN GOSUB530:GOSUB510:
GOTO740

400 IFD$="2"THENGOSUB570:GOSUB530:G
OTO760

410 IFD$="3"THENGOSUB510:GOSUB570:G

OTO780

420 GOTO350

430 PRINT:PRINT"1- P=V*I"

440 PRINT"2- P=I↑2*R"

450 PRINT"3- P=V↑2/R"

460 INPUTE$

465 PRINT:PRINT

470 IF E$="1"THEN GOSUB550:GOSUB530
:GOTO810

480 IFE$="2"THEN GOSUB530:GOSUB510:
GOTO830

490 IFE$="3"THEN GOSUB550:GOSUB510:
GOTO850

500 GOTO430

510 INPUT"RESISTENCIA EN OHM";R

520 RETURN

530 INPUT"INTENSIDAD EN AMPER";I

540 RETURN

550 INPUT"TENSION EN VOLT";V

560 RETURN

570 INPUT"POTENCIA EN WAT";P

580 RETURN

590 REM CALCULOS PARA EL PARAMETRO
RESISTENCIA

600 PRINT:PRINT"R=";V;"/";I;"=";V/I
;" OHM"

610 GOSUB880:GOTO910

620 PRINT:PRINT"R=";P;"/";I;"↑2=";P
/(I↑2);" OHM"

630 GOSUB880:GOTO940

640 PRINT:PRINT"R=";V;"↑2/";P;"=";V
↑2/P;" OHM"

650 GOSUB880:GOTO970

660 REM CALCULOS PARA EL PARAMETRO

INTENSIDAD

670 PRINT:PRINT"I=";V;"/";R;"=";V/R
;" AMPER"

680 GOSUB880:GOTO1000

690 PRINT:PRINT"I=";P;"/";V;"=";P/V
;" AMPER"

700 GOSUB880:GOTO1030

710 PRINT:PRINT"I=SQR(";P;"/";R;")=
";SQR(P/R);" AMPER"

720 GOSUB880:GOTO1060

730 REM CALCULOS PARA EL PARAMETRO
TENSION

740 PRINT:PRINT"V=";I;"*";R;"=";I*R
;" VOLT"

750 GOSUB880:GOTO1090

760 PRINT:PRINT"V=";P;"/";I;"=";P/I
;" VOLT"

770 GOSUB880:GOTO1120

780 PRINT:PRINT"V=";R;"*SQR(";P;"/"
;R;")=";R*SQR(P/R);" VOLT"

790 GOSUB880:GOTO1150

800 REM CALCULOS PARA EL PARAMETRO
POTENCIA

810 PRINT:PRINT"P=";V;"*";I;"=";V*I
;" WATT"

820 GOSUB880:GOTO1180

830 PRINT:PRINT"P=";I;"↑2*";R;"=";I
↑2*R;" WATT"

840 GOSUB880:GOTO1210

850 PRINT:PRINT"P=";V;"↑2/";R;"=";V
↑2/R;" WATT"

860 GOSUB880:GOTO1240

870 REM CALCULOS DEL CUARTO PARAMET
RO

880 PRINT:PRINT"DESEA SABER EL VALO
R  DEL 4[COMMY]'PARAMETRO?"

890 INPUT"S/N";G$

900 RETURN

904 IFG$="N"THEN1270

906 RETURN

910 IFG$="S"THENPRINT:PRINT"P=";V*I
;" WATT":GOTO1270

920 GOSUB904:GOTO610

940 IFG$="S"THENPRINT:PRINT"V=";P*I
;" VOLT":GOTO1270

950 GOSUB904:GOTO630

970 IFG$="S"THENPRINT:PRINT"I=";P/V
;" AMPER":GOTO1270

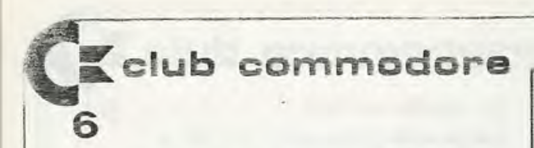
980 GOSUB904:GOTO650

1000 IFG$="S"THENPRINT:PRINT"P=";V↑
2/R;" WATT":GOTO1270

1010 GOSUB904:GOTO680

1030 IFG$="S"THENPRINT:PRINT"R=";V↑
2/P;" OHM":GOTO1270
```

(continua en la pag. siguiente)



COLABORACIONES

PROGRAMA PARA EL ESTUDIO DE LA LEY DE OHM

(continuación)

```
1040 GOSUB904:GOTO700

1060 IFG$="S"THENPRINT:PRINT"V=";R*
SQR(P/R);" VOLT":GOTO1270

1070 GOSUB904:GOTO720

1090 IFG$="S"THENPRINT:PRINT"P=";I↑
2*R;" WATT":GOTO1270

1100 GOSUB904:GOTO750

1120 IFG$="S"THENPRINT:PRINT"R=";P/
(I↑2);" OHM":GOTO1270

1130 GOSUB904:GOTO770

1150 IFG$="S"THENPRINT:PRINT"I=";SQ
R(P/R);" AMPER":GOTO1270

1160 GOSUB904:GOTO790

1180 IFG$="S"THENPRINT:PRINT"R=";V/
I;" OHM":GOTO1270

1190 GOSUB904:GOTO820

1210 IFG$="S"THENPRINT:PRINT"V=";I*
R;" VOLT":GOTO1270

1220 GOSUB904:GOTO840

1240 IFG$="S"THENPRINT:PRINT"I=";V/
R;" AMPER":GOTO1270

1250 GOSUB904:GOTO860

1270 PRINT:PRINT"DESEA REALIZAR ALG
UN  CALCULO MAS?"

1280 INPUT"S/N";H$

1290 IFH$="S"THENPRINT"[ CLR ]":GOT
070

1300 IFH$="N"THENEND

1310 GOTO1270

READY.
```

Nuestra segunda colaboración consiste en un sencillo programa para niños, del cual su autor nos dice:

Este programa es un juguete para mi hijo de tres años de edad, que cuando tengo la maldita ocurrencia de cargarlo en memoria, me impide tocar una sola tecla en lo que queda de tarde.

Para los que quieran jugar con el tren de mi hijo, un aviso: una vez en marcha, ya no se podrá listar; sólo se podrá jugar o apagar el VIC.

Para los avispados: ¡BUSCAD LA INSTRUCCIÓN DE BLOQUEO!

Por nuestra parte añadiremos que para mover el tren deben pulsarse las teclas: V para marcha adelante; X para marcha atrás; C cambia el color del tren y la barra espaciadora hace sonar el silbato.

```
READY.

1 REM TREN.F.CASTANYER

10 POKE36879,8:POKE37150,2:POKE788,
194:A=12

20 PRINT"[ CLR ][ YEL ][RVSON][CRSRD]
[CRSRR][CRSRR][CRSRR][CRSRR][CRSRR]
[CRSRR][CRSRR][CRSRR][CRSRR]TREN"

30 PRINT"[CRSRD][CRSRD][CRSRR][ GRN ]
PER NINS DE 3-5 ANYS"

40 PRINT"[CRSRD][CRSRD][CRSRD][CRSRR]
[CRSRR][SHIFZ]MARXA A DRETA > V"

50 PRINT"[CRSRD][CRSRD][CRSRR][CRSRR]
[SHIFZ]MARXA  ENRERA > X"

60 PRINT"[CRSRD][CRSRD][CRSRR][CRSRR]
[SHIFZ]CANVIAR COLOR > C"

70 PRINT"[CRSRD][CRSRD][CRSRD][CRSRR]
SIULET > BARRA ESPAI[CRSRD][CRSRD]"
:PRINT"    PULSA [RVSON][ RED ]RETU
RN"

75 IFPEEK(197)<>15THEN75

80 A$="[ WHT ][ RED ][ CYN ][ PUR ]
[ GRN ][ BLU ][ YEL ]"

90 FORC=1TO5:N=INT(RND(1)*7)+1:A$(C
)=MID$(A$,N,1):NEXT

100 PRINT"[ CLR ][CRSRD][CRSRD][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD]"

110 PRINTA$(1)"   [COMMO][COMMO][COMMI]
[COMMO][COMMI][COMMO][COMMO] [COMMO]
[COMMI][COMMO][COMMI]"

120 PRINTA$(2)"    [COMM+] [COMM+]
[COMM+]   [COMM+] [COMM+]   [COMM+]
"

130 PRINT"    [COMM+] [COMM+] [COMM+]
   [COMM+][ WHT ]B"A$(2)"[COMM+] [COMM+]
 [COMM+]"

140 PRINT"    [COMM+][COMM+][COMM+]
[COMM+][COMM+]   [COMM+][COMM+][COMM+]

[COMM+][COMM+][COMM+][COMM+]"

150 PRINTA$(3)"   [COMM+][COMM+][COMM+]
[COMM+][COMM+][COMM+][COMM+][COMMO]
[COMMO][COMM+][COMM+][COMM+][COMM+]
[COMM+][COMM+][COMM+][COMM£]"

160 PRINT"   [COMM+] [COMM+] [COMM+]
 [COMM+] [COMM£][COMM+] [COMM+] [COMM+]
 [COMM+]"

170 PRINTA$(4)"     [SHIFQ] [SHIFQ]
     [SHIFQ] [SHIFQ] [SHIFQ] "

180 PRINTA$(5)"[COMMU][COMMU][COMMU]
[COMMU][COMMU][COMMU][COMMU][COMMU]
[COMMU][COMMU][COMMU][COMMU][COMMU]
[COMMU][COMMU][COMMU][COMMU][COMMU]
[COMMU][COMMU][COMMU][COMMU]"

190 POKE36878,15

200 FORC=0TO88STEP44:POKE7874-C,32:
NEXT

205 IFPEEK(197)=27THENA=A+1:IFA>23T
HENA=1

210 IFPEEK(197)=26THENA=A-1:IFA<1TH
ENA=23

220 IFPEEK(197)=32THENPOKE36874,225
:POKE36876,230

230 FORC=0TO88STEP44:POKE7874-C,81:
NEXT

240 IFPEEK(197)=26ORPEEK(197)=27THE
NPOKE36877,200+A

250 IFPEEK(197)<>32THENPOKE36874,0:
POKE36876,0

260 IFPEEK(197)=34THEN90

270 IFPEEK(197)=15THENPOKE36864,12:
GOTO10

280 POKE36864,A:POKE36877,0:GOTO200

READY.
```

«MÉTODO DE SIFTING»

Comparado con los métodos lineales descritos hasta ahora, se diferencia de ellos en que no se mantiene secuencia de comparación fija. Sin embargo, como veréis más adelante, no deja de ser una burbuja más evolucionada que la cocktelera (si no habéis leído los números anteriores, esto os puede sonar a chino).

El «Sifting« actúa igual que el método de la burbuja hasta que se produce un intercambio. Entonces el elemento que se ha comparado con el de clave menor se le fuerza a subir tan arriba como se pueda. Se va comparando hacia atrás con sus elementos anteriores y efectuando intercambios hasta que se encuentre un elemento de clave. Entonces el proceso normal de arriba a abajo continúa desde la posición en que se efectuó el intercambio inicial. La clasificación termina cuando se llega al elemento **N**, intentándolo comparar con el **N+1**, en el proceso de arriba a abajo.

Este método resulta muy fácil de programar y da como resultado un programa relativamente corto, teniendo en cuenta los excelentes resultados que da. Sin lugar a dudas es mejor que los métodos lineales vistos hasta el momento.

La rutina de «Sifting» desarrollada en BASIC se muestra en el listado número 1.

Con este método hemos llegado al final de los métodos lineales, en los que cada elemento se compara con el anterior o el siguiente.

En la figura 1 aparece un cuadro resumen de estos métodos.

MÉTODOS NO LINEALES

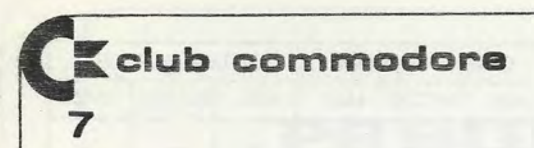
Dedicaremos esta última parte del artículo a describir un método no lineal para finalizar esta serie dedicada a las técnicas de ordenación.

En los métodos no lineales, como su nombre indica, el recorrido de la tabla no se hace de forma lineal sino que los elementos pueden dar saltos a través de ella. Del método de cálculo de estos saltos dependerá la eficacia de la rutina.

«MÉTODO DE SHELL»

Este método une a una gran potencia y rapidez su brevedad, si bien no es fácil de programar a pesar de su sencilla base teórica.

(sigue el texto en la pag. 8)



SOTWARE DE BASE (III)



métodos de Sifting y de Shell

por **E. MARTÍNEZ DE CARVAJAL**

LISTADO 1

```
READY

10 REM *** SIFTING ***
20 REM -----------------------------------
30 REM METODO DE ORDENACION
40 REM
50 REM ERNESTO MTNZ. DE CARVAJAL HE
DRICH
60 REM
70 REM 26-FEB-83
80 REM -----------------------------------
100 INPUT "NUMERO ELEMENTOS";M9
110 DIM W(M9),O(M9)
120 FOR I=1 TO M9
130 W(I)=INT(RND(0)*900+100):O(I)=W
(I)
140 NEXT I
150 FOR I=1TOM9:PRINTW(I):NEXTI
200 REM *** RUTINA DE ORDENACION **
*
220 TI$="000000"
280 J=M9-1
310 FOR I=1 TO J
320 IF W(I)<=W(I+1) THEN 350
330 A1=W(I)
340 W(I)=W(I+1):W(I+1)=A1:J3=I
345 GOSUB 380
350 NEXTI
355 GOTO 440
380 FOR K=J3 TO 2 STEP-1
390 IFW(K)>=W(K-1) THEN K=2:GOTO420
400 A1=W(K)
410 W(K)=W(K-1):W(K-1)=A1
420 NEXT K
430 RETURN
440 REM FIN ORDENACION
3500 T$=TI$
3510 OPEN1,4
3520 PRINT#1,"ORDENACION DE  UNA TA
BLA DE "M9" ELEMENTOS POR EL METODO
 DE SIFTING"
3530 PRINT#1,"TIEMPO DE ORDENACION
: "LEFT$(T$,2)":"MID$(T$,3,2)":"RIG
HT$(T$,2)
3540 PRINT#1
3550 PRINT#1,"TABLA INICIAL  TABLA
FINAL"
3560 PRINT#1,"[SHIFR][SHIFR][SHIFR]
[SHIFR][SHIFR][SHIFR][SHIFR][SHIFR]
[SHIFR][SHIFR][SHIFR][SHIFR][SHIFR]
  [SHIFR][SHIFR][SHIFR][SHIFR][SHIFR]
[SHIFR][SHIFR][SHIFR][SHIFR][SHIFR]
[SHIFR]"
3570 FOR I=1TOM9
3580 PRINT#1,"     "O(I)"        "W
(I)
3590 NEXT I
4000 PRINT#1
4010 CLOSE1
4020 STOP
READY.
```

métodos de Sifting y de Shell

(conclusión)

MÉTODO	NECESIDADES DE ESPACIO	COMPARACIONES			INTERCAMBIOS		
		MÍN.	MEDIO	MÁX.	MÍN.	MEDIO	MÁX.
ENUMERACIÓN	N a 2 N	$\frac{N^2}{2}$	$\frac{N^2}{2}$	$\frac{N^2}{2}$	N	—	—
SELECCIÓN	N a 2 N	N^2	N^2	N^2	N	—	—
BURBUJA	N	N	$\frac{N^2}{2}-\frac{3N}{4}$	$\frac{N^2}{2}$	0	$\frac{N^2}{4}$	$\frac{N^2}{2}$
SIFTING	N	$\frac{N^2}{2}$	$\frac{N^2}{2}$	$\frac{N^2}{2}$	N	—	—

Fig. 1

Este método lo considero ideal para micro-ordenadores, ya que es una rutina muy rápida en tablas de hasta 10.000 (diez mil) elementos, lo cual es más que suficiente en la mayoría de las aplicaciones y además se programa en ocho (¡ocho!) líneas en BASIC.

La idea del método, publicado por primera vez por Donald L. Shell en 1959, es tratar de superar la desventaja que supone para el «método de Sifting» el que los elementos de la lista sólo se puedan mover a través de la misma posición a posición, lo cual — inevitablemente — da tiempos de ejecución proporcionales a **N** ^ 2. Por tanto, para mejorarlo habría que introducir algún mecanismo mediante el cual los elementos puedan dar largos saltos a través de la tabla. Esto es precisamente lo que hace este método, dando tiempos de ordenación proporcionales a **N** ^ (5/4) lo cual es sensiblemente menor a **N** ^ 2. Me perdonaréis que no os detalle más este método pero es que puedo morir en el intento (y creo que todavía no hay medallas al mérito informático). El listado 2 reproduce la rutina que espero que os sea de mucha utilidad.

LISTADO 2

```
READY.

10 REM *** SHELL ***

20 REM -----------------------------
-----

30 REM METODO DE ORDENACION

40 REM

50 REM ERNESTO MTNZ. DE CARVAJAL HE
DRICH

60 REM

70 REM 26-FEB-83

80 REM -----------------------------
-----

100 INPUT "NUMERO ELEMENTOS";M9

110 DIM A4(M9),O(M9)

120 FOR I=1 TO M9

130 A4(I)=INT(RND(0)*900+100):O(I)=
A4(I)

140 NEXT I

200 FOR I=1TOM9:PRINTA4(I):NEXTI

300 TI$="000000"

3000 REM *** RUTINA DE ORDENACION *
**

3030 REM DATOS A PASAR

3040 REM A4(I)=MATRIZ A ORDENAR

3050 REM M9  =NUMERO DE ELEMNTOS A
ORDENAR

3080 I=0:K=0

3090 I=3*I+1:K=K+1:S0(K)=I:IFI<M9TH
EN3090

3095 K=K-3:IFK<1THENK=1

3100 FORI=KTO1STEP-1:S=S0(I):FORJ=1
TOS:FORI0=J+STOM9STEPS

3110 V=A4(I0):FORJ0=I0TOJ+SSTEP-S

3120 IFV>=A4(J0-S)THENA4(J0)=V:GOTO
3140

3130 A4(J0)=A4(J0-S):NEXTJ0:A4(J)=V

3140 NEXT I0:NEXTJ:NEXTI:REM FIN OR
DENACION

3500 T$=TI$

3510 OPEN 1,4

3520 PRINT#1,"ORDENACION DE UNA TAB
LA DE "M9" ELEMENTOS POR EL METODO
SHELL"

3530 PRINT#1,"TIEMPO DE ORDENACION
: "LEFT$(T$,2)":"MID$(T$,3,2)":"RIG
HT$(T$,2)

3540 PRINT#1

3550 PRINT#1,"TABLA INICIAL  TABLA
FINAL"

3560 PRINT#1,"[SHIF*][SHIF*][SHIF*]
[SHIF*][SHIF*][SHIF*][SHIF*][SHIF*]
[SHIF*][SHIF*][SHIF*][SHIF*][SHIF*]
  [SHIF*][SHIF*][SHIF*][SHIF*][SHIF*]
[SHIF*][SHIF*][SHIF*][SHIF*][SHIF*]
[SHIF*]"

3570 FOR I=1TOM9

3580 PRINT#1,"     "O(I)"        "A
4(I)

3590 NEXT I

4000 PRINT#1

4010 CLOSE1

4020 STOP

READY.
```

Los tiempos resultantes para ordenaciones de 50 y 100 elementos en los dos métodos descritos en este artículo son:

Método	50 elementos	100 elementos
Sifting	00:00:16	00:01:00
Shell	00:00:06	00:00:19

■

PROGRAMAS 

PROGRAMAS EN DISCO

D-1001 Agenda 5.000
Permite contener 4 páginas de información de 114 personas, entidades, etc. Pudiendo añadir, cambiar o borrar la información que se desee.
En castellano, necesita ampliación de 8K. Presentado en caja.

D-1002 QSL 3.000
Registro de contactos para radioaficionados, versión en disco.
Necesita ampliación de 8K. Presentado en caja.

D-1003 Test Demo 3.000
Programa de test para la unidad de disco. Nota: este disco es el que se suministra con la unidad de disco VIC-1540.

D-1004 Assembler 5.000
Editor y compilador en BASIC para rutinas en lenguaje máquina del 6502. Necesita ampliación de 3K. Acompañado de manual en inglés.

D-1005 English language 2.500
Versión en disco del programa en cinta, dado que al manejar ficheros resulta mucho más rápida su utilización.
Necesita ampliación de 8K. Presentado en caja.

D-1006 Quiz Master 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

D-1007 Matemáticas 1 (nivel BUP) 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

3301 Simplicalc 13.000
Hoja de Trabajo Electrónica 60 columnas por 100 filas Max. Necesita ampliación de 16K.

3304 Vic File 13.000
Bases de datos para el VIC-20 con posibilidad de definir el formato de los campos. Necesita ampliación de 16K. Ideal para fichero de clientes.

3305 Vic Writer 13.000
Proceso de texto muy potente y rápido. Necesita ampliación de 8K ó 16 K.

PROGRAMAS EN CINTA

Programas Técnicos

C-128 Programación lineal 1.000
Método simplex. Cálculo del valor de las variables que satisfaciendo las restricciones hacen máxima o mínima una función. En castellano. Presentado en caja.

C-129 Matrices 1.000
Suma, resta, multiplicación, multiplicación por un escalar e inversión de matrices. En castellano. Presentado en caja.

C-131 Regresiones I 1.000
Contenido: Regresión lineal: Cálculo por mínimos cuadrados de la recta que se ajusta mejor a una nube de puntos.
Regresión múltiple: Variable dependiente en función de N variables independientes de grado 1. En castellano. Presentado en caja.

C-132 Regresiones II 1.000
Contenido: Regresión de orden N: Variable dependiente en función de variable independiente de grado N.
Regresión exponencial: Ajuste nube de puntos a una curva exponencial.
Regresión geométrica: Ajuste de una nube de puntos a una curva geométrica. En castellano. Presentado en caja.

C-133 Estadística I 1.000
Distribución normal, Poisson, binomial, chi-cuadrado, Student, F de Snedecor. En castellano. Presentado en caja.

C-134 Estadística II 1.000
Contenido: Cálculo de la media, varianza y desviación tipo, tanto de la muestra como de la población, estando los datos agrupados o no. Test de chi-cuadrado y test de Student. En castellano. Presentado en caja.

C-135 Sistemas 1.000
Contenido: Resolución de sistemas de N ecuaciones con N incógnitas.
Resolución de ecuaciones de grado 2 dando las soluciones tanto reales como complejas.
Cálculo de permutaciones y de combinaciones. En castellano. Presentado en caja.

C-137 Integración 1.000
Cálculo de integrales por el método de Gauss.
Cálculo de derivadas de una función en un punto. Interpolaciones tanto lineales como curvilíneas. En castellano. Presentado en caja.

Programas Educativos

C-125 Hangmath 1.500
Se trata de averiguar la multiplicación entre 2 números (tanto los números como la multiplicación y el resultado final), procurando no ser ahorcado. Las instrucciones son en castellano.

C-140 Skymath 1.500
Sumas y restas para chicos muy jóvenes saliendo naves espaciales para motivarlos. Una manera de aprender jugando. 3K.

C-141 Space Division 1.500
20 preguntas sobre divisiones muy sencillas. En función de los resultados correctos, despega un cohete que llegará más o menos lejos. 3K.

C-143 English Language 2.000
Especialmente adecuado para las personas que, poseedoras de un buen nivel de inglés, quieren alcanzar cotas superiores. Necesita ampliación de 8K. Presentado en caja y acompañado de su correspondiente manual.

C-145 Mastermind 3.000
Participe en el famoso concurso de la BBC, de preguntas y respuestas sobre temas variados.
Este cassette va acompañado del sistema operativo y de un file de demostración. Necesita ampliación de 8K. Presentado en caja.

C-146 Matemáticas I (nivel BUP) 2.000
Contiene sistema operativo y 7 cuestionarios sobre diferentes temas abarcando geometría, aritmética, matemáticas generales, álgebra, etc. Necesita ampliación de 8K ó 16K. Presentado en caja.

PROGRAMAS COMERCIALIZADOS POR "MICROELECTRÓNICA Y CONTROL, S.A."
Puede encontrarlos en su distribuidor más próximo

microelectrónica y control s a 



El mejor ordenador personal del Mundo



El COMMODORE 64

Este es el nuevo ordenador personal COMMODORE 64. Un gigante de 40 cm. con un precio casi tan pequeño como su tamaño.

Nadie hasta ahora había logrado ofrecerle 64 K de memoria, 40 columnas en pantalla, 8 sprites y un sonido de auténtica maravilla por sólo 110.000,— ptas. Claro que tampoco todo el mundo es el líder mundial en microordenadores.

COMMODORE sabe perfectamente que para seguir siendo el número uno, tiene que estar constantemente en vanguardia. De calidad. De precios. De todo. Para ello investigamos constantemente.

Afortunadamente nuestra labor se ve plenamente recompensada cuando vemos, como lo demuestra el cuadro comparativo, que nuestro más directo competidor cuesta nada menos que un 100% más caro. Y ello sin reunir todos los adelantos técnicos del COMMODORE 64.

---

1. Capacidad total de memoria RAM de 64 K. Interpretador BASIC extendido y sistema operativo residentes en ROM.

2. Dotado del más potente chip sintetizador de sonido diseñado hasta hoy, el COMMODORE 64 ofrece 3 voces totalmente independientes con una gama de 9 octavas. El programa puede controlar la envolvente, la afinación y la forma de onda de cada voz.

convirtiendo al COMMODORE 64 en el mejor simulador de instrumentos.

3. Conectable directamente a toda una gama de periféricos, incluyendo unidad de discos, impresora de matriz de puntos o de margarita, plotter, comunicaciones locales y remotas.... y mucho más.

4. Pantalla de alta resolución en color con 320 × 200 puntos directamente direccionables. Capacidad en modo carácter de 25 líneas por 40 columnas.

5. El chip de video, único en su género, permite el uso de 8 «Sprites» (figuras móviles en alta resolución y color). Los «Sprites» pueden moverse independientemente por programa de pixel» en «pixel».

6. A cada «Sprite» se le asigna por programa un nivel de prioridad en caso de cruce con otro, consiguiendo efectos tridimensionales, existiendo también detección automática de colisiones.

7. Teclado profesional con mayúsculas y minúsculas, más 62 caracteres gráficos, todos ellos disponibles en el teclado y visualizables en 16 colores, en forma normal o bien en video invertido.

8. Encontrará a su disposición una completa gama de programas profesionales, incluyendo proceso de textos, sistemas de información, modelos financieros, contabilidad y muchas más aplicaciones.

9. Están en fase de desarrollo asimismo otros lenguajes tales como LOGO, UCSD PASCAL, COMAL, ASSEMBLER, etc. Todos los programas existentes de la gama COMMODORE, desde el VIC-20 hasta los modelos CBM pueden ser adaptados fácilmente.

10. Posibilidad de inserción de cartuchos con programas grabados en ROM, tanto profesionales como para educación y ocio.

11. Opción de un segundo procesador Z-80 para trabajar con sistema operativo CP/M (R).

EL COMMODORE 64 Y SU MAS DIRECTO COMPETIDOR

OPCION DE BASE	COMMODORE 64	Más directo competidor
Precio	110.000.— ptas.	El doble
Memoria usuario	64 K	48 K
Teclado profesional	SI	SI
Teclado con caracteres gráficos	SI	NO
Minúsculas	SI	NO
Teclas de función	SI	NO
Máxima capacidad disco	170 K a 1 M	143 K
AUDIO		
Generador de sonido	SI	SI
Sintetizador de música	SI	NO
Salida HI-FI	SI	NO
VIDEO		
Salida monitor	SI	SI
Salida para TV	SI	EXTRA
PERIFERICOS		
Cassette	SI	SI
Periféricos inteligentes	SI	SI
Bus serie	SI	NO
SOFTWARE		
Opción CP/M (R)	SI	SI
Ranura cartucho externo	SI	NO

commodore
COMPUTER

PARA MAS INFORMACION
DEL COMMODORE 64,
LLAMAR O ESCRIBIR A:
MICROELECTRONICA Y CONTROL
c/ Taquígrafo Serra, 7, 5º. Barcelona-29
Tel. (93) 250 51 03
c/ Princesa, 47, 3º, G. Madrid-8
Tel. (91) 248 95 70

Nombre
Dirección
Tel.
Población

SLOGAN

PROGRAMAS 

Programas Aplicaciones

C-130 **Caja** 2.000
Dentro de una cuenta bancaria permite abrir hasta 100 subcuentas con su correspondiente debe y haber. Preparado para funcionar si se desea con impresora, unidad de discos y evidentemente cassette. Necesita ampliación 16K. Presentado en caja.

C-136 **Dieta** 2.000
En función de la edad, sexo, altura, configuración y actividad desempeñada por la persona, da el peso ideal. Pregunta si se desea seguir un régimen, y en función de la actividad física desarrollada da el peso que se debe tener y la pérdida diaria que se debe alcanzar hasta llegar a dicho peso óptimo. Está en castellano y necesita ampliación de 8K. Presentado en caja.

C-139 **Vicalc** 1.500
El VIC-20 se comporta como una calculadora y utiliza la notación polaca inversa. Manual de instrucciones en castellano.

C-142A **Interface de RTTY y CW y cassette de CW** . 25.000
Especial para radioaficionados.

C-142B **Cassette de RTTY** 2.500
Para utilizarlo es necesario el interface de RTTY y CW. Especial radioaficionados.

C-144 **Quiz Master** 2.000
Programa para preparación de cuestionarios. Especialmente adecuado para escuelas. En castellano. Necesita ampliación de 8K.

PROGRAMAS EN CARTUCHO

1901 **Avenger** 4.500
Típica batalla de marcianitos. Sus enemigos avanzan en fila mientras disparan teniendo como protecciones su rapidez y velocidad de disparo y unos cuantos bunkers.

1902 **Star Battle** 4.500
Batalla de marcianitos con ataque de ovnis estilo «stukas». Se juega con teclado o con Joystick.

1904 **Super Slot** 4.500
Juego en cartucho que le permitirá jugar con una máquina tragaperras cómodamente desde su casa y frente al televisor. Se juega por teclado o con Joystick.

1906 **Super Allien** 4.500
Juego en cartucho. El individuo que usted maneja deberá construir defensas en el laberinto frente al ataque de sus enemigos que le pretenderán devorar.

1907 **Júpiter Lander** 4.500
Este juego en cartucho le permitirá demostrar su habilidad al intentar aterrizar en Júpiter en condiciones muy difíciles y con el peligro de que el combustible se le agote. Se juega por teclado.

1908 **Draw Poker** 4.500
Si quiere jugar al poker contra un enemigo implacable e impasible, este cartucho se lo permitirá. Se juega por teclado.

1909 **Road Race** 4.500
Con este cartucho podrá usted conducir un coche a través de una autopista. Se juega por cartucho.

1910 **Radar Ratrace** 4.500
Conduce un ratón a través de un laberinto procurando comer el queso y evitar que sea cazado por los gatos o por otros ratones que están al acecho. Mediante un radar sabe en qué posición se encuentran tanto sus enemigos como su objetivo: el queso. Se juega con Joystick o teclado.

1911 **The sky is failling** 4.500
Tiene que evitar a toda costa que el muro alcance su altura. Se juega con paddle.

1912 **Mole attack** 4.500
Debe darle con un martillo a un topo que va apareciendo por agujeros diversos. Sumará puntos positivos o negativos en función de si le da en la cabeza o en la base posterior.

1919 **Sargon II chess** 4.500
Juego de ajedrez con 7 niveles de dificultad. A partir del nivel 1 existe la opción de que el VIC-20 le indique qué jugada haría si estuviese en su lugar, por lo que le hace adecuado para los que se inicializan en este juego. Se juega por teclado o Joystick.

1924 **Omega Race** 4.500
Con este cartucho podrá pasar un rato divertido simulando batallas espaciales. Se juega con joystick.

C-400 **VIC Forth** . . . . consultar
Lenguaje de alto nivel.

C-401 **VIC Stat** . . . . consultar
Paquete de utilidades estadísticas.

C-402 **VIC Graph** . . . . consultar
Resolución de funciones por métodos gráficos.

C-403 **Wordcraft** 42.500
Compatible Series CBM 8000 y CBM 4000. Tratamiento de textos.

C-404 **VIC Screen Master** 10.000
Paquete de utilidades para edición de pantallas con utilidades para manejo de cadenas y disco.

BIBLIOGRAFÍA

Manual usuario VIC-20 800
Nociones de funcionamiento del VIC-20 y del lenguaje BASIC.

Curso Introducción BASIC: Parte I 2.500
De una manera clara y sencilla, partiendo de cero, se va enseñando el lenguaje del futuro. Este libro va acompañado de dos cassettes con 17 programas en castellano.

Curso Introducción BASIC: Parte II 2.500
Profundización de lo tratado en Parte I, especialmente en creación de ficheros, manejo de subrutinas, efectos musicales, aspectos animados, etc.

VIC Revealed 1.500
Libro que explica con detalle el funcionamiento interno del VIC-20. Incluye esquemas. Libro fotocopiado en inglés.

Programers Reference Guide 2.000
Profundización en el manejo del VIC-20. Incluye referencias a rutinas lenguaje máquina. Libro fotocopiado en inglés.

PROGRAMAS COMERCIALIZADOS POR "MICROELECTRÓNICA Y CONTROL, S.A."
Puede encontrarlos en su distribuidor más próximo

microelectrónica y control s a 



TRUCOS (II)

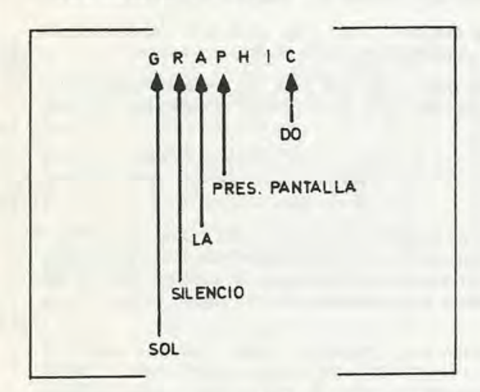


superexpander

por **R. PARDO**

Seguramente sabréis que pulsando simultáneamente CTRL y ⇐ se entra en el modo música del cartucho, pero, ¿habéis probado alguna vez pulsar una tecla programable dentro de este modo? Si no lo habéis hecho nunca, ¡probadlo! Oiréis que suenan unas notas y aparecen unas letras en pantalla. ¿Por qué?

Ved esto en la tecla F1:



¡Luego puedo «programar» mi melodía favorita dentro de una cadena alfanumérica en el modo música sin programa y pulsando sólo una tecla! La única limitación son los 82 caracteres máximos que se puede introducir de texto en KEY. Pero, ¡es igual!, nos quedan siete teclas más a programar, lo que nos da la friolera (introduciendo sólo notas, sin cambio alguno) de ¡656 notas diferentes! Impresionante, ¿verdad? También diré que se puede programar el modo música dentro de un programa. ¿Cómo? Probad esto:

10 PRINT «f»

donde «f» es una F con inversión de color (resultado de pulsar simultáneamente CTRL y ⇐ [flecha a la izquierda]). Lo único que queda es entrar cadenas alfanuméricas en sentencias PRINT y el VIC generará música. ¿Veis qué fácil?

NOTA: Para los que posean los cartuchos «Ayuda Programador» o «Superexpander» con el Curso de Introducción al BASIC parte I ó II, debéis saber que con los programas «PIANO» y «TECLADO» se pueden utilizar las teclas de función (con alguno de los dos cartuchos conectados) para hacerlos servir de caja de música, con un sonido algo más sofisticado de lo que da de por sí solo el «Superexpander». De la manera anteriormente descrita podemos programar con las teclas de función nuestra melodía favorita, entrando las teclas que se hubieran de haber pulsado manualmente. Si en el momento de la programación está conectado el cartucho «Superexpander» se dispondrá de 656 notas como máximo a programar. Y si se cuenta con «Programmer's Aid», se dispondrá de 984 tonos diferentes. Y como precaución adicional la programación de las teclas de función deberá realizarse **antes de ejecutar el programa musical (piano o teclado).** Fijaos, seguramente habréis leído el micro-manual que se entrega con el cartucho y habréis visto los comandos de trazado, de pintado, etc., pero seguramente (como me pasó a mí) os preguntaréis, ¿y con todo esto, cómo programo? Seguid estas instrucciones:

Paso 1: Seleccionar el escalón de resolución deseado, entre los cinco que hay para escoger, a saber:
0-modo standard del VIC (22*23)
1-modo multicolor (80*160)
2-modo HI-RES (160*160)
3-modos HI-RES y multicolor mezclados
4-retorno al modo texto (modo normal)
Luego escribid: 10 GRAPHIC 2

Paso 2: Definición de los colores a utilizar. No es estrictamente necesario, pero no está de más hacerlo. Así pues, suponiendo que «definimos», podéis escribir:

20 COLOR 0,1,2,3

Donde la pantalla tomará el color negro (0), el marco el color blanco (1), el carácter el color rojo (2), y el color auxiliar el color cyan (3).

Paso 3: Selección de la función de trazado, pintado o de impresión de carácter que se quiera ejecutar: podemos trazar una recta (DRAW...TO), un círculo o una elipse (CIRCLE), un punto (POINT), pintar un área (PAINT) o cambiar de color de carácter (REGION).

La pantalla se organiza en una pseudo-matriz de 1024*1024 puntos, pero en realidad la máxima resolución que se alcanza con el «Superexpander» es de 160*160 puntos.

Los comandos de «Superexpander» trabajan bien con todos los comandos BASIC, excepto con IF...THEN. Si el comando del «Superexpander» debe seguir la palabra THEN, éstas deben estar separadas de la siguiente manera:

IFX=1THEN:SCNCLR

Ahora bien, un programa escrito con comandos del «Superexpander» sólo funcionará bien cuando esté el cartucho conectado y consecuentemente, los programas que estén escritos con comandos del cartucho sin que éste esté conectado NO serán reconocidos por éste cuando se ejecute el programa, ya que no encontrará los «tokens» que éste asigna a sus comandos. ■

MEA CULPA

UNA MEJORA SIGNIFICATIVA DEL PROGRAMA PARA HACER LISTADOS. — En la pasada edición de CLUB COMMODORE publicamos, bajo el mismo título que encabeza esta nota, una mejora de nuestro programa de listados; pero nuestro entrañable enemigo — BUG — nos jugó una de sus típicas malas pasadas en la línea 165. La versión correcta es la siguiente:

```
120 GOSUB 165
165 IFASC(A$)>31ANDASC(A$)<144
    THENRETURN
167 IFASC(A$)>223THENRETURN
```

MULTICENTRO DE INFORMATICA



PUERTO RICO, 21-23 - MADRID-16 TEL. 250 74 02 - 250 74 04

commodore VIC-20

JUEGOS

- COMECOCOS. 3,5K. A/R. G/E. JY. EXTRAORDINARIA VERSION DEL POPULAR PUCKMAN. COLOR Y MOVILIDAD INSUPERABLE . 1.900
- VICGAMON. +3K. JUEGO DE INTELIGENCIA QUE LE MANTENDRA EN TENSION HASTA DERROTAR A SU VIC 1.800
- ASTEROIDS WAR. 3,5K. C/M. A/R. G/E. JY. ESPECTACULAR BATALLA GALACTICA CONTRA LA NUBE PROTONICA EN 3 DIMENSIONES 1.800
- FROGGER. +3K y 3,5K. C/M. A/R. G/E. JY. ULTIMA NOVEDAD EN EE.UU. CRUZAR EL PELIGROSO RIO Y LA AUTOPISTA SUICIDA 2.000
- RATMAN. +8K. C/M. A/R. G/E. DE LA BOVEDA CELESTE DESCENDERAN EXTRAÑAS RATAS ATOMICAS. ESPECTACULAR ANIMACION 1.900
- SHARK ATTACK. 3,5K. C/M. A/R. JY. EN MEDIO DEL OCEANO SERA ATACADO POR LOS PELIGROSOS TIBURONES. DEFIENDASE CON SU RED ATOMICA 1.900
- ROX III. 3,5K Y+8K. C/M.A/R. G/E. JY. DESDE SU SOFISTICADA BASE LUNAR DEFIENDA SU PLANETA DEL ATAQUE DE LOS UFOS 1.800
- ,ULTISOUND SYNTHETIZER. 3,5K. ¿UN ORGANO EN SU VIC? ¿CON ACOMPAÑAMIENTO, BATERIA Y EFECTOS ESPECIALES? .. 1.900
- SKI-RUN. 3,5K. C/M. A/R. G/E. DESLICESE POR LAS HELADAS PISTAS DE COMPETICION. SLALOM, S/GIGANTE, DESCENSO. 9 NIVELES 1.800
- FIREBIRD. (SPACE PHREEKS). 3,5K. C/M. A/R. G/E. JY. AÑO 3.010. VD. ES EL UNICO SUPERVIVIENTE DE LA BATALLA DE RIGELLIAN. DEBERA COLONIZAR OTRO PLANETA Y LUCHAR CONTRA LAS CRIATURAS GALACTICAS 1.900
- BREAKOUT. 3,5K. CONSIGA DESTRUIR LA PARED DE LADRILLOS MULTICOLORES CON LA BOLA MAGICA. INCLUYE "MASTERMIND". 1.600
- AJEDREZ. PRIMERA VERSION EN CASSETTE CON GRAFICOS EN ALTA RESOLUCION. BASTANTES NIVELES DE JUEGO. (STANDARD) . 2.800
- SHADOWFAX. INCREIBLES GRAFICOS ANIMADOS. EL CABALLERO DE LAS SOMBRAS EN LUCHA CONTRA LOS JINETES DEL TIRANO INVASOR. (STANDARD) 1.900
- SNAKE. COLORIDO, MOVIMIENTOS Y GRAFICOS EXCEPCIONALES. VERSION DEL FAMOSO JUEGO DE LAS SERPIENTES (SNAKE). (STANDARD) 1.900
- MYRIAD. +3K. C/M. A/R. G/E. JY. LA MAS ESPECTACULAR AERONAVE PARA DESTRUIR LAS CRIATURAS COSMICAS EN SU VIAJE GALACTICO 2.000
- COSMIADS. 3,5K. C/M. A/R. G/E. JY. VERSION ULTRARRAPIDA DEL MUNDIALMENTE FAMOSO JUEGO "GALAXIANS". INCREIBLES EFECTOS SONOROS 1.700
- BLITZRIEG. 3,5K. C/M. A/R. G/E. JY. DESTRUYA LA CIUDAD ENEMIGA CON SU BOMBARDERO. 25 NIVELES DE JUEGO 1.600
- DEFENSA. +8K. C/M. A/R. G/E. JY. N.º 1 EN INGLATERRA, COMO GUERRERO GALACTICO DEBE DEFENDER A LOS HUMANOIDES CONTRA LOS ENEMIGOS CIBERNETICOS. 9 NIVELES DE JUEGO. ESPECTACULAR NAVE Y SONIDOS 2.000
- VIC PANIC. 3,5K. C/M. A/R. G/E. JY. VERSION DEL POPULAR "SPACE PANIC". ESCALE LAS LADERAS Y HUYA DE LOS MONSTRUOS 1.800
- SKRAMBLE. 3,5K. C/M. A/R. G/E. JY. ATRAVESANDO LOS TEMIBLES PASADIZOS INTERESTELARES DESTRUYA LAS BASES ENEMIGAS 1.900
- 3D LABYRINTH. +8K. C/M. A/R. EXTRAORDINARIO LABERINTO TRIDIMENSIONAL. ¿SERA CAPAZ DE SALIR DE EL? UNO O VARIOS JUGADORES 1.800
- GOLF. 3,5K. RECORRIDO DE 9 HOYOS PERO ATENCION A LOS OBSTACULOS: ARBOLES, LAGOS, ETC. INCLUYE VIC MUSIC Y PIANO 1.600
- CARRERA DE BUGGYS. 3,5K. C/M. A/R. G/E. ESPECTACULAR RECORRIDO. ACELERADOR. DECELERACION. 9 NIVELES 1.800
- GRIDRUNNER. 3,5K. C/M. A/R. G/E. JY. IMPRESIONANTE VERSION LLENA DE COLORIDO, MOVILIDAD Y SONIDO DEL POPULAR "CENTIPEDE" 1.900
- HI-RES. 3,5K. GRAN JUEGO DEMO/UTILIDAD PARA REALIZAR EN PANTALLA GRAFICOS EN ALTA RESOLUCION. INCLUYE GEN. CARACTERES 1.500
- ABDUCTOR. LAS CRIATURAS COSMICAS DEL PLANETA "ALPHA I" INTENTARAN SECUESTRAR A LOS HUMANOIDES PARA CONSEGUIR ENERGIA E INTELIGENCIA SUPERIORES. TU MISION SERA DEFENDER TU PLANETA Y DESTRUIR LAS NAVES ABDUCTORAS. (STANDARD) 1.800
- TRAXX. VERSION DEL CONOCIDO JUEGO "AMIDAR": MEZCLA DEL POPULAR "PACKMAN" Y DEL JUEGO "QUIX". 100% CODIGO MAQUINA. GRAFICOS EN ALTA RESOLUCION. ESPECTACULAR SONIDO Y COLOR. 8K DE MEMORIA 2.000

UTILIDADES

- VIC PRINT. +8K. EXTRAORDINARIO Y SENCILLO PROCESADOR DE TEXTOS. TABULACION, MAQUETACION, CABECERAS, COPIAS. CASS O DISK 2.000
- VIC LABEL. +8K. EN COMBINACION CON VIC PRINT. ELABORA ETIQUETAS PARA DIRECCIONES 1.900
- VIC POST. +8K. ELABORA LETRAS Y TEXTOS ESPECIALES EN TAMAÑO Y FORMA PARA POSTERS, LISTAS DE PRECIOS, ETC. .. 2.900
- VIC CALC. HERRAMIENTA DE CALCULO QUE SUSTITUYE AL LAPIZ, PAPEL Y CALCULADORA. REALIZA COMPLEJOS MODELOS FINANCIEROS CON POSIBILIDAD DE AJUSTARLO A OTROS PARAMETROS CON SOLO PULSAR UNA TECLA. 16K DE MEMORIA 3.200
- VIC BASE. 16K. POTENTE BANCO DE DATOS. 255 CARACTERES, MAS DE 25 CAMPOS. CAMBIO Y LOCALIZACION. SALIDA IMPRESORA 3.200
  OPCION 3,5K 1.800
- GRAPHVICS. +3K. AÑADE 18 POTENTES COMANDOS PARA POSICIONAR PUNTOS, DIBUJAR LINEAS Y TEXTOS EN ALTA RESOLUCION (152×160) 2.200
- GRAPH EDITOR & SOFTKEY 24. 3,5K. AMBOS PROGRAMAS PERMITEN DISEÑAR HASTA 64 CARACTERES PARA INCORPORARLOS A SUS PROPIOS PROGRAMAS Y JUEGOS 2.000

EDUCATIVOS

- QUIZ-MASTER. +3K. EL MAS ESPECTACULAR AVANCE EDUCATIVO. PERMITE LA CORRECCION Y PUNTUACION DE TODAS LAS RESPUESTAS QUE RECIBE EL ORDENADOR ....
- QUIZ SET-UP. EN TANDEM CON QUIZ—MASTER PERMITE LA ELABORACION POR EL USUARIO DE TODO TIPO DE PREGUNTAS Y CUESTIONES EDUCATIVAS O DE ENTRETENIMIENTO. EGB, IDIOMAS, MATEMATICAS, HISTORIA, GEOGRAFIA, ETC. CREANDO UN AGIL Y ATRACTIVO SISTEMA DOMESTICO/EDUCATIVO ..... } 3.200
- FACEMAKER. 16K. CARICATURANDO EL ROSTRO DE SUS COMPAÑEROS Y AMIGOS EL VIC 20 PONDRA A PRUEBA EL VOCABULARIO Y LA ATENCION DEL NIÑO 2.000
- NUMBER CHASER. 16K. PROGRAMA PARA PRACTICAS DE MULTIPLICACION CON CARRERAS DE COCHES. ADELANTA, FRENA, ACELERA SEGUN LAS RESPUESTAS. 4 NIVELES DE DIFICULTAD ... 2.000
- NUMBER GULPER. 16K. JUEGO EDUCACIONAL DE COMPETICION CON NUMEROS PARA SUMA, RESTA, MULTIPLICACION Y DIVISION 2.000
- WE WANT TO COUNT. 16K PROGRAMA PARA NIÑOS A PARTIR DE TRES AÑOS, INVASORES, CARRERAS, ETC. 2.000
- TWISTER. 16K. JUEGO DE LOGICA Y CONCENTRACION. PUZZLES GEOMETRICOS CON SONIDO Y COLOR 2.000

LIBROS

VIC REVEALED	2.200	ASSEMBLER	2.000	ZAP! POW! BOOM!	1.800
GETTIN ACQUAINTED WITH YOUR VIC 20	1.800	SYNPHONY MELANCHOLY COMP.	1.800	VIC INNOVATIVE	2.000
50 PROGRAMAS LISTADOS I	1.500	50 PROGRAMAS LISTADOS II	1.500	50 PROGRAMAS LISTADOS III	1.500

APLICACIONES EDUCATIVAS



editor personal
(VIC-20 con ampliación de 3 K o sin ampliación)

por **JOAN ARGEMÍ**

Joan Argemí es un usuario de VIC que trabaja en aplicaciones de enseñanza. Hemos conseguido convencerle para que nos escriba una sección de nuestra Revista dedicada a estos temas que creemos son de gran interés para nuestros lectores. Aunque este primer artículo cae un poco fuera del título de esta sección creemos que vale la pena pues permite aprovechar toda la superficie de la pantalla del televisor y mediante las pertinentes modificaciones —que improvisará el lector habilidoso— se puede convertir en un interesante (y barato) procesador de textos.

El programa que os presentamos tiene su aplicación más directa en la confección (secreta) de un diario personal. Con él podréis escribir, día a día, vuestras ideas, pensamientos, poesías..., sin el temor de que sean directamente legibles por algún familiar desaprensivo. Podréis también recordar en él los hechos más significativos de un día concreto, como citas o notas...; y todo ello guardado en un cassette secreto, cassette que, para ser leído, necesitará la llave maestra de este programa. Es casi imposible que con estos métodos de seguridad, alguien se inmiscuya en vuestra intimidad. El programa ofrece la posibilidad pues, de escribir amplios textos en la pantalla (hasta 26 columnas por 32 líneas), y luego grabarlos en cinta o bien imprimirlos.

Si os fijáis en el listado, veréis que el programa se inicia con el planteamiento de una variable que concierne al registro del VIC (línea 1). Manipulando con ella, podremos variar a nuestra conveniencia la situación vertical y horizontal de la parte útil de la pantalla. Para ello (muchos de vosotros lo sabréis de memoria), sólo debéis cambiar el contenido de los POKEs de las líneas 6 (izquierda - derecha, entre los números 9 y 20) y 7 (arriba - abajo, entre los números 10 y 40). El programa los sitúa en 10 y 30, respectivamente, pero éstos son solamente los contenidos convenientes a la televisión de su autor... Para vosotros pueden ser distintos. En las líneas 2 y 3 el programa propone las variables VR (vertical), es decir número de columnas y HR (horizontal), es decir número de líneas. Como podéis ver en el listado, el programa da a estas variables unos contenidos de 26 y 28 (respectivamente), pero se pueden variar a vuestro gusto, mientras no excedan de 26 por 32.

Las líneas 15, 16 y 17 definen a las rutinas de borrar (BO), grabar (GR), y cargar (CR), todas ellas contenidas en los data del final del listado, codificadas como valores del POKE de la línea 19.

El programa se maneja con las teclas normales para todos los asuntos de edición (Rvs,Cursor,Del), y con tres teclas de función para los modos principales. Así, con la F1, el programa grabará todo lo escrito en la pantalla (a tener en cuenta, el hecho de que no aparecerán mensajes y desaparecerá el cursor). Con la tecla F3, el programa podrá cargar cualquier escrito previamente grabado por él, y, con la F5, el programa imprimirá en formato pantalla todo lo que en ella esté escrito. Si quisiéseis cambiar el formato de lo impreso, sólo deberéis sustituir a las variables VR y HR de la línea 56 por los números del formato que de-

(continua en la pag. siguiente)

editor personal

(viene de la pag. anterior)

seéis. Así pues, sólo os queda picar cuidadosamente el listado, proponerle el número de columnas y líneas más conveniente a vuestro televisor y... a inspirarse.

Si queréis ser perfeccionistas en adaptar vuestro programa al formato exacto con que lo hagáis, cambiar los 26*32 de las líneas 38 y 39 por vuestros VR*HR pues estos valores están puestos como límite máximo, y si ejercéis en formatos menores, el cursor recorre espacios inexistentes, que tampoco es tan molesto... pero el VIC es para perfeccionistas.

Desde ahora podréis facilitar la tarea a vuestros futuros biógrafos, guardando celosamente en una cassettoteca vuestras impresiones de la vida... ¡Ah!, pero no se os olvide dejarles en un sobre lacrado el secreto programa con que se puedan leer. ■

LISTADO "EDITOR PERSONAL"

```
1 VIC=36864

2 VR=26

3 HR=28

4 DEF FNA(I)=I-128*(I<128)

5 DEF FNB(I)=I+128*(I>127)

6 POKE VIC,10

7 POKE VIC+1,30

8 POKE VIC+2,PEEK(VIC+2) AND 128 OR
 VR

9 POKE VIC+3,PEEK(VIC+3) AND 129 OR
 HR*2

10 POKE 36866,PEEK(36866) AND NOT 1
28

11 REM

12 SC=7168:CO=37888

13 POKE 648,28

14 MC=SC-256

15 BO=MC

16 GR=MC+35

17 CR=MC+66

18 POKE 56,27:POKE 52,27

19 FORI=0TO85:READ A:POKE MC+I,A:NE
XT

20 SYS MC

21 P=SC

22 RV=0

23 POKE P,FNA(PEEK(P)):FORI=1 TO 40

24 IF PEEK(198)<>0 THEN POKEP,FNB(P
EEK(P)):GOTO27

25 NEXT:POKE P,FNB(PEEK(P)):FORI=1T

040:IF PEEK(198)<>0 THEN 27

26 NEXT:GOTO 23

27 A$="":GET A$

28 IF A$="[ CLR ]" THEN SYS BO:P=SC
:GOTO23

29 IF A$="[HOME ]"THEN P=SC:GOTO 23

30 IF A$="[RVSON]" THEN RV=1:GOTO 2
3

31 IF A$="[RVSOF]" THEN RV=0:GOTO23

32 IF A$="[CRSRD]" THEN P=P+VR

33 IF A$="[CRSRU]" THEN P=P-VR

34 IF A$="[CRSRR]" THEN P=P+1

35 IF A$="[CRSRL]" THEN P=P-1

36 IF A$=CHR$(13) THEN P=SC+VR+VR*I
NT((P-SC)/VR)

37 IF A$=CHR$(20) THEN POKE P,32:P=
P-1

38 IF P<SC THEN P=P+(26*32)

39 IF P>=SC+(26*32) THEN P=P-(26*32
)

40 IF A$="[CRSRD]" OR A$="[CRSRU]"
OR A$="[CRSRR]" OR A$="[CRSRL]" THE
N 23

41 IF A$=CHR$(20) OR A$=CHR$(13) TH
EN 23

42 IF A$="[  F1 ]" THEN GOSUB 64:GO
TO 23

43 IF A$="[  F3 ]" THEN GOSUB 65

44 IF A$="[  F5 ]" THEN GOSUB 54:GO
TO 23

45 REM

46 A=ASC(A$):IFA>31 AND A<64 THEN G
OTO 51

47 IF A>63 AND A<96 OR A>160 AND A<
192 THEN A=A-64:GOTO 51

48 IF A=255 THEN A=94:GOTO 51

49 IF A>191 AND A<224 THEN A=A-128:
GOTO 51

50 GOTO 23

51 IF RV=1 THEN A=FNA(A)

52 POKE P,A:IF P<SC+HR*VR-1 THEN P=
P+1

53 GOTO 23

54 A$=CHR$(145):IF (PEEK(36869) AND
 2)=2 THEN A$=CHR$(17)

55 OPEN 4,4:IF A$=CHR$(17) THEN CLO
SE 4:OPEN 4,4,7

56 PRINT#4:A=SC-VR:FOR B=0 TO HR-1:
B$=A$:A=A+VR:FOR C=A TO A+VR-1

57 D=PEEK(C):IF D>128 THEN D=D-128:
RV=1:B$=B$+CHR$(18)

58 IF D<32 THEN D=D+64:GOTO 62

59 IF (D>31) AND (D<64) THEN 62

60 IF(D>63) AND (D<96) THEN D=D+32:
GOTO 62

61 IF (D>95) AND (D<128) THEN D=D+6
4

62 B$=B$+CHR$(D):IF RV=1 THEN B$=B$
+CHR$(146):RV=0

63 NEXT:PRINT#4,B$:NEXT:PRINT#4:CLO
SE4:RETURN

64 POKE 648,30:PRINT"[HOME ][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD][CRSRD]
[CRSRD]":SYS GR:POKE 648,28:RETURN

65 POKE 648,30:PRINT"[HOME ][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD][CRSRD]
[CRSRD][CRSRD][CRSRD][CRSRD]":SYS C
R:POKE 648,28:RETURN

66 DATA 162,0,169,32,157,0,28,157,0
,29

67 DATA 157,0,30,157,0,31,169,0,157
,0

68 DATA 148,157,0,149,157,0,150,157
,0

69 DATA 151,232,208,225,96,234,162,
1

70 DATA 160,255,32,186,255

71 DATA 169,0,32,189,255,169,28,133
,2,169,0,133,1

72 DATA 169,1,162,64,160,31,32,216,
255,96,234

73 DATA 162,1,160,255,32,186,255,16
9,0,32,189

74 DATA 255,162,255,160,255,32,213,
255,96,234

READY.
```

NOTICIAS

novedades COMMODORE 64

Acaban de aparecer los primeros programas para el Ordenador Personal COMMODORE 64. Vamos a ver, brevemente, algunas de las características más sobresalientes de ellos para información de nuestros lectores.

El primero es un Procesador de Textos llamado EASY SCRIPT para la confección de cartas, circulares, manuales y — cómo no — artículos como el que usted está leyendo. Veamos cuáles son sus cualidades:

— Dispone de la mayoría de ayudas para la edición (modificación de palabras, correcciones, etc...) de los procesadores de textos, siendo éstas muy sencillas tanto de recordar como de ejecutar.

— **EASY SCRIPT** tiene la posibilidad de trabajar en **MODO REPORT,** con lo cual facilita muchísimo la confección de informes con datos extraídos de ficheros.

— **EASY SCRIPT** puede trabajar con un máximo de 240 columnas tanto en pantalla como en impresora. En modo editor, todos los comandos programados para ejecución son visibles para el operador.

— El comando de salida permite visualizar el documento en su forma final, tanto en impresora como en pantalla, facilitando en gran manera la corrección de los posibles errores, o bien imperfecciones de forma, antes de efectuar la impresión definitiva.

— **EASY SCRIPT** permite, mediante una función especial, realizar cualquier comando de disco. Con lo cual pueden visualizarse directorios, borrarse ficheros, cambiar un nombre, etc. **SIN SALIR DEL PROPIO EASY SCRIPT** y sin perder un solo dato de memoria.

— **EASY SCRIPT** puede mandar caracteres especiales a la impresora, como por ejemplo funciones ESCAPE seguidas de cualquier número de caracteres.

— Dispone además de **BACK SPACE** considerado en la justificación del texto, con lo cual queda asegurada la posibilidad de los diversos acentos y caracteres especiales.

— Con las impresoras **COMMODORE** puede trabajar con caracteres definibles por el usuario.

— Finalmente, la facilidad de aprendizaje de los potentes comandos de los cuales va provisto el programa, hacen de **EASY SCRIPT** el mejor aliado de cualquier profesional para la realización fácil y práctica de sus trabajos.

El programa se presenta en disco. El precio de venta al público es de 20.000 pesetas.

* * *

CALC RESULT es un nuevo paquete de software para oficinas, creado por la firma HANDIC SOFTWARE AB.

Para las personas que han de realizar cálculos o simulaciones complejos, **CALC RESULT** les resultará un instrumento valiosísimo, que no sólo acelerará el trabajo sino que simplificará en gran medida las tareas complejas.

CALC RESULT permite configurar hasta 32 páginas electrónicas con un máximo de 154 filas y 63 columnas, a la manera de tablas, donde almacenar cantidades y textos para interaccionar entre ellas y realizar operaciones de simulación numérica del tipo «¿qué pasaría si...?» u operaciones repetitivas complejas.

CALC RESULT incorpora un gran número de comandos y funciones que aceleran la manipulación de datos.

CALC RESULT opera de modo tridimensional, proporcionando una visión natural del trabajo. Esta estructura tridimensional facilita la creación de esquemas, amplía las posibilidades de manipulación de datos y suma diferentes páginas electrónicas de información.

Las intersecciones entre hileras y columnas crean miles de posiciones, también llamadas coordenadas. En cada una de estas posiciones, el usuario puede entrar números, textos o fórmulas de cálculo. Los comandos de formatación permiten configurar cada posición con un formato diferente.

Cada una de las páginas **CALC RESULT** puede ser escindida horizontal y verticalmente, lo que permite ver simultáneamente dos diferentes partes de una página.

(continua en la pag. siguiente)

novedades COMMODORE 64

(viene de la pág. anterior)

Es posible colocar diferentes páginas electrónicas en cada mitad de la pantalla escindida. Una ventanilla puede abrirse en la pantalla, de modo que muestre cualquier punto de la página, y puede escindirse en dos mitades, al igual que la propia pantalla. De este modo es posible poner en la pantalla simultáneamente partes de cuatro páginas diferentes.

Para simplificar el uso, **CALC RESULT** visualiza todas las opciones que ofrece cada comando y si se desea carga un menú de ayuda en castellano con una breve explicación de cada comando.

Las funciones de edición de **CALC RESULT** permiten cambiar, insertar y borrar texto, fórmulas y valores. La información entrada puede reorganizarse en cualquier ocasión deseada, para entrar nuevas hileras y columnas.

Después de entrar una fórmula matemática, la misma puede repetirse tantas veces como se desee en cualquier hilera o columna. **CALC RESULT** suma, calcula promedios y manipula completas hileras de columnas o áreas de posiciones. **CALC RESULT,** al contrario de otros programas de hoja electrónica desplazable, interpreta las fórmulas con arreglo a las reglas normales de aritmética.

CALC RESULT retiene en su memoria interna las fórmulas que usted escribe, incluso después de ejecutar el cálculo. De esta forma, cada vez que usted cambia algún valor, todos los demás valores que están relacionados con él se recalculan automáticamente. Este recálculo, enteramente automático o de desearse manual, es un eficaz instrumento de planificación y cálculo, que permite no sólo corregir errores, sino también estudiar diferentes predicciones y soluciones.

CALC RESULT permite reproducir los valores también como histogramas de barras, con escalas a definir por el usuario, adaptadas a la magnitud de los valores. Estas tablas gráficas pueden entonces formatearse individualmente para su impresión.

CALC RESULT se presenta en cartucho acompañado de un disco y su precio es de 33.000 pesetas.

* * *

FORTH es un nuevo lenguaje estructurado que permite al usuario una gran flexibilidad al poder definir las propias funciones incluyendo rutinas en código máquina (contiene el equivalente a un MACRO ASSEMBLER) para aquellas partes del programa que requieran un tratamiento especial por razones de tiempo de ejecución y/o flexibilidad, con lo que en un determinado momento se puede llegar a trabajar con un lenguaje «a medida» de la aplicación que se está desarrollando. Además, por razón de su estructura interna, reúne las ventajas de un lenguaje compilado (rapidez) y uno interpretado (utilización eficaz de los recursos de memoria del equipo).

Su presentación es en cartucho con manual en inglés, y su precio es de 20.000 pesetas. ■

NOTICIAS

adaptación de la impresora S-30 de Honeywell al VIC-20

Microelectrónica y Control, S. A. anuncia la disponibilidad de la impresora de tipo profesional S-30 de Honeywell conectable al VIC-20 mediante un interfaz especial que se suministra con ella, siendo las características de esta impresora las siguientes:

— Impresión por matriz de puntos (7×7) controlada por microprocesador.

— Arrastre de papel: tipo tractor, ancho entre 3 y 15 pulgadas. Alimentación trasera o inferior.

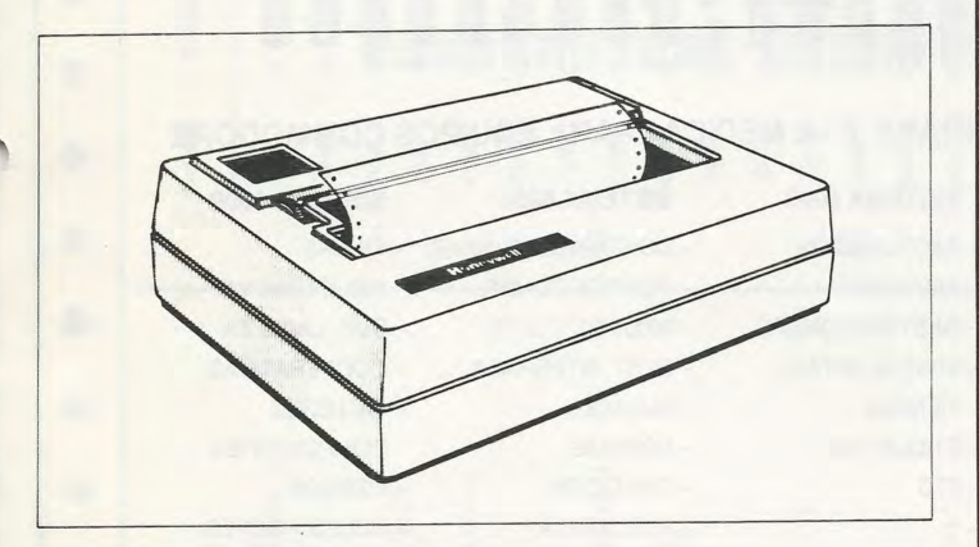
— Hasta un original y dos copias, máximo grosor: 0.3 milímetros.

— Auto test.

— Buffer de línea.

— Cubierta anti-ruido.

— Avance de línea: 100 mseg.



— Velocidad de impresión: 80 caracteres por segundo, bidireccional. 50 LPM. para líneas de 80 caracteres.

— Set de 96 caracteres.

— 132 caracteres/línea, 10 caracteres/pulgada, 6 líneas/pulgada.

— Cartucho de cinta tintadora estandar Honeywell, en bucle Moebius que permite el máximo aprovechamiento.

— Su precio de venta al público es de 110.000 pesetas. ■

TRUCOS

valor correcto para los bytes libres en el COMMODORE 64

por **P. MASATS**

Una de las primeras sorpresas que obtuvimos del COMMODORE 64 fue el comprobar la memoria libre usuario de que disponíamos al poner en marcha el primer equipo que cayó en nuestras manos (operación muy frecuente como es lógico). Al hacer ?FRE(0) la respuesta fue un número negativo, lo que causó no poco estupor y buen número de incisivos e irónicos comentarios entre los presentes. Bien, la explicación es sencilla: la lógica interna de la función FRE(X) da el resultado como un número entero (ver el artículo VENTANA CBM; ESTRUCTURA DE LAS CABECERAS DE VARIABLES Y ARRAYS; CLUB COMMODORE, número 6). El formato de un número entero está formado por dos Bytes en los cuales el bit más significativo del Byte más significativo contiene el signo del número, siendo igual a uno cuando la cantidad representada es negativa. Cuando el intérprete de BASIC cuenta el número de Bytes libres en la rutina de inicialización que se ejecuta en la puesta en marcha, el recuento (al exceder la máxima capacidad que es 32767) da como resultado la puesta a uno de este bit y la rutina de impresión exhibe un número negativo. (Esperamos que no haya muchas desmoralizaciones prematuras después de esta «sencilla» explicación). La manera de evitar comportamientos excéntricos de la función FRE(X) en un programa que la utilice es hacer, por ejemplo:

```
X=FRE (0) : IF X<0 THEN
X=X+65536
```

En este caso la variable X contendrá el valor correcto de Bytes libres. ■

MARKETCLUB

- En Barcelona, clases de informática. PLAZAS LIMITADAS. Lenguaje BASIC. Prácticas con micro-ordenador VIC-20. Prof. E. Martínez de Carvajal. Información: Tel. 345 10 00. Srta. María José (mañanas) o 345 87 75. Sr. Martínez (fuera de horas de oficina).
- Vendo módulo para VIC-20 compuesto de: dos puertas de ocho entradas/salidas más dos de control y estado. Dos temporizadores programables. Posibilidad de interrupción IRQ-NMI. Con las correspondientes instrucciones. Razón: Luis Torrents. C/. Velázquez, 39. TERRASSA (Barcelona).
- Hago programas en BASIC COMMODORE bajo encargo. Vendo VIC-20, con 16 K de RAM, cartuchos de: Ayuda al Programador, Monitor Lenguaje Máquina, Superexpander + 3 K y tres juegos. Además se darán al comprador muchos programas hechos y comprados y 5 libros existentes o no en España sobre el VIC. Todo por 65.000 ptas. (vale más de 100.000). Por separado también. Llamar al 91 - 253 13 40. Horas comida y cena. Dirección: Francisco Gutiérrez. Santiago Rusiñol, 12. MADRID-3.
- Vendo aplicación de facturación con control de representantes, 9 listados, 6 ficheros, estadísticas, etc. Permite copias de seguridad. Configuración: VIC 8 K, disco e impresora, 40.000 pesetas. Escribir a Jaime Ameller Pons. General Mola, 15, 1º B. CALATAYUD (Zaragoza).
- Se ofrece 3008 + C2N de segunda mano en buenas condiciones; precio a convenir. Razón: Domingo Garrofé Trabal. Aragón, 386, 1º 1ª. Tel. 211 54 40. BARCELONA-9.
- Vendo interfaz y programa para RTTY y CW para el PET a 15 K. Rafael, EA3CGK. Avenida Barcelona, 21, A, 4º 2ª. IGUALADA (Barcelona).

COMMODORE 64 - MAPA DE MEMORIA 4 / ABREVIATURAS DE LOS COMANDOS BASIC EN EL COMMODORE 64

```
FFFA; Hardware vectors
EF59; RS232 receive
EF7E; Setup to receive
EFC5; Receive parity error
EFCA; Receive overflow
EFCD; Receive break
EFD0; Framing error
EFE1; Submit to RS232
F00D; No-DSR error
F017; Send to RS232 buffer
F04D; Input from RS232
F086; Get from RS232
F0A4; Check serial bus idle
F0BD; Messages
F12B; Print if direct
F13E; Get..
F14E;     ..from RS232
F157; Input
F199; Get.. tape/serial/rs232
F1CA; Output..
F1DD;     ..to tape
F20E; Set input device
F250; Set output device
F291; Close file
F30F; Find file
F31F; Set file values
F32F; Abort all files
F333; Restore default I/O
F34A; Do file open
F3D5; Send SA
F409; Open RS232
F49E; Load program
F5AF; °searching°
F5C1; Print filename
F5D2; °loading/verifying°
F5DD; Save program
F68F; Print °saving°
F69B; Bump clock
F6BC; Log PIA key reading
F6DD; Get time
F6E4; Set time
F6ED; Check stop key
F6FB; Output error messages
F72D; Find any tape headr
F76A; Write tape header
F7D0; Get buffer address
F7D7; Set buffer start/end pointers
F7EA; Find specific header
F80D; Bump tape pointer
F817; °press play..°
F82E; Check tape status
F838; °press record..°
F841; Initiate tape read
F864; Initiate tape write
F875; Common tape code
```

Comando	Abreviatura	Aspecto en la pantalla
ABS	A SHIFT B	A
AND	A SHIFT N	A
ASC	A SHIFT S	A
ATN	A SHIFT T	A
CHR$	C SHIFT H	C
CLOSE	CL SHIFT O	CL
CLR	C SHIFT L	C
CMD	C SHIFT M	C
CONT	C SHIFT O	C
COS	NONE	COS
DATA	D SHIFT A	D
DEF	D SHIFT E	D
DIM	D SHIFT I	D
END	E SHIFT N	E
EXP	E SHIFT X	E
FN	NONE	FN
FOR	F SHIFT O	F
FRE	F SHIFT R	F
GET	G SHIFT E	G
GET#	NONE	GET#
GOSUB	GO SHIFT S	GO
GOTO	G SHIFT O	G
IF	NONE	IF
INPUT	NONE	INPUT

Comando	Abreviatura	Aspecto en la pantalla
INPUT#	I SHIFT N	I
INT	NONE	INT
LEFT$	LE SHIFT F	LE
LEN	NONE	LEN
LET	L SHIFT E	L
LIST	L SHIFT I	L
LOAD	L SHIFT O	L
LOG	NONE	LOG
MID$	M SHIFT I	M
NEW	NONE	NEW
NEXT	N SHIFT E	N
NOT	N SHIFT O	N
ON	NONE	ON
OPEN	O SHIFT P	O
OR	NONE	OR
PEEK	P SHIFT E	P
POKE	P SHIFT O	P
POS	NONE	POS
PRINT	?	?
PRINT#	P SHIFT R	P
READ	R SHIFT E	R
REM	NONE	REM
RESTORE	RE SHIFT S	RE
RETURN	RE SHIFT T	RE

Comando	Abreviatura	Aspecto en la pantalla
RIGHT$	R SHIFT I	R
RND	R SHIFT N	R
RUN	R SHIFT U	R
SAVE	S SHIFT A	S
SGN	S SHIFT G	S
SIN	S SHIFT I	S
SPC(	S SHIFT P	S
SQR	S SHIFT Q	S
STATUS	ST	ST
STEP	ST SHIFT E	ST
STOP	S SHIFT T	S
STR$	ST SHIFT R	ST
SYS	S SHIFT Y	S
TAB(	T SHIFT A	T
TAN	NONE	TAN
THEN	T SHIFT H	T
TIME	TI	TI
TIME$	TI$	TI$
USR	U SHIFT S	U
VAL	V SHIFT A	V
VERIFY	V SHIFT E	V
WAIT	W SHIFT A	W